

2003 FAA National Software Conference

Software Development Tools



Assessment of Software Development Tools for Safety Critical Real-Time Systems

FAA Contract DTFA0301C00048

Faculty: **A.J.Kornecki**, N.Brixius, J.Zalewski (FGCU)

Students: J.P.Linardon, J.Labbe, K. Hall, H. Lau, L. Crawford,
C. Sanouillet, S. Lakha, J. Poole, D. Hearn, T. Osako

Department of Computer and Software Engineering

Embry Riddle Aeronautical University

Daytona Beach, FL 32114

phone: (386) 226-6888

kornecka@erau.edu

<http://faculty.erau.edu/korn>

Presentation at the FAA National Software Conference

September 16-19, 2003,

Reno, NV



ERAU research conducted for the FAA under contract DTFA0301C00048

page 1



Outline

- FAA Viewpoint
- Research Plan
- Development Tool Categories
- Survey - Industry Feedback
- Qualification and Determinism
- Assessment and the Taxonomy
- Current Work
- Conclusion
- Appendix: Development Tool Landscape



ERAU research conducted for the FAA under contract DTFA0301C00048

page 2

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Introduction: DO-178B Definitions

- The terms defined and referenced in DO-178B section 12.2 (expanded in N8110.49):
 - **Software tool:** *A computer program used to help develop, test, analyse, produce or modify another program or its documentation*
 - **Software development tools:** *Tools whose output is part of airborne software and thus can introduce errors*
- These documents define criteria and data needed for tools qualification to establish confidence that the tools are dependable enough to certify the system developed with their support
- The specific interpretation is left to the applicant and the certifying authorities



ERAU research conducted for the FAA under contract DTFA0301C00048

page 3



Real Time Safety

Introduction: Research Need

- Tools improve development process by automation and reduction of repetitive tasks
- Qualified tools could help with the current certification process by reducing the verification burden of the intermediate software lifecycle artifacts
- Although tool qualification is a well established concept, in practice only few commercial tools were attempted to be qualified:
 - Requirements for qualification of development tools seem to be overly restrictive
 - Economics of the industry has been preventing tool vendors to engage in development tools qualification process



ERAU research conducted for the FAA under contract DTFA0301C00048

page 4

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Research Plan: Objectives

FAA Contract DTFA0301C00048

■ Objectives:

- To establish a base for assessment of software development tools
- To create a taxonomy and a set of criteria/guidelines for tool selection and qualification
- To perform an experiment collecting data using selected development tools

NOTE: the investigation focus on the tools which are in demand considering the status of technology and the industry needs

■ Three-phase research activity carried out with the students and faculty of the ERAU software engineering program (January 2002 - December 2004)

- Phase One: Baseline and Taxonomy
- Phase Two: Experiment and Feedback
- Phase Three: Assessment and Guidelines



ERAU research conducted for the FAA under contract DTFA0301C00048

page 5



Real Time Safety

Research Plan

■ Questions:

- What are the categories of development tools?
- What kinds of development tools are we using today?
- What are the basic capabilities of a development tool?
- What kinds of development tools do we anticipate using in the future?
- Why do we need to qualify tools?
- What tools are we considering for qualification?
- What tools were attempted to be qualified?
- How to achieve qualification for COTS tools?
- What are barriers to qualification of development tools
- What factors need to be addressed regarding development tools and qualification?
- What would help to encourage safe usage of development tools?
- What would help to enable tool qualification?

■ Methodology:

data collection, data processing, data synthesis, infrastructure decisions, and data integration



ERAU research conducted for the FAA under contract DTFA0301C00048

page 6

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Research Plan: Problem Statement

■ Qualification and Industry View

- What tools have been attempted to be qualified to DO-178B standard? What are the basic concepts regarding software tool use in the regulated field of safety critical software development? What is the industry opinion on the current tool qualification process? What tool qualification approaches meet safety needs and are acceptable to both industry and certifying authorities?

■ Quality Assessment

- How may the quality of a software development tool from the perspective of its use in safety critical real-time system development be assessed? What are the mechanisms and methods for evaluating software development tool quality? What evaluation criteria should be used?

■ Tool Evaluation Taxonomy

- What are the functionalities of modern software development tools? How can the tools be categorized? Which categories and functions are vital for the development process? Which categories and functions of software development tools need to be evaluated?



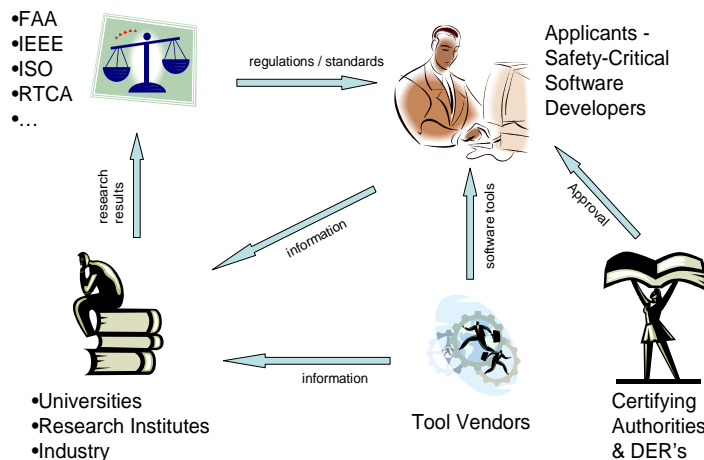
ERAU research conducted for the FAA under contract DTFA0301C00048

page 7



Real Time Safety

Software Aspects of Certification Stakeholder Diagram




ERAU research conducted for the FAA under contract DTFA0301C00048

page 8

2003 FAA National Software Conference


Software Development Tools



Real Time Safety


Software Development Tool Categories

- Software development tools categories (DO-178B):
 - Requirements
 - Design
 - Coding
 - Integration
- The current practice of software engineering software development tools categories:
 - Analysis Tools (requirements, performance)
 - Design Tools (creation, reuse)
 - Implementation Tools (build, run-time support)
- All development tools define a functional **transformation** between the **input** and **output** artifacts depending on system **state**



ERAU research conducted for the FAA under contract DTFA0301C00048


page 9



Real Time Safety

Software Development Tool Categories: Functionality and Transformations

- Once the requirements are developed and verified, the tools in **design phase** will facilitate orderly and correct translation of the requirements into the executable code
 - **Design Tool:** requirements -> design diagrams
 - **Scheduling Tool:** timing information -> schedule
 - **Code Generator:** design diagram -> source code
 - **Compiler:** source code -> object code
 - **Documentation Tool:** text /diagrams -> documents
- Design tools can be based on **structural** approach (OO/UML) or **functional** approach (block/dataflow)
- Both approaches use some form of state machines to represent system dynamics

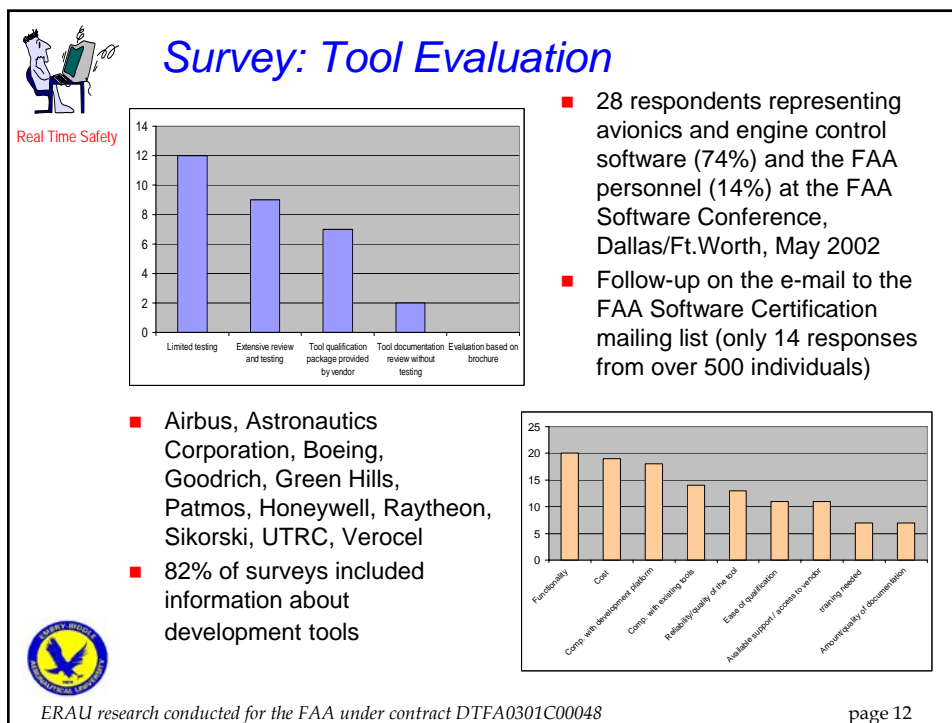
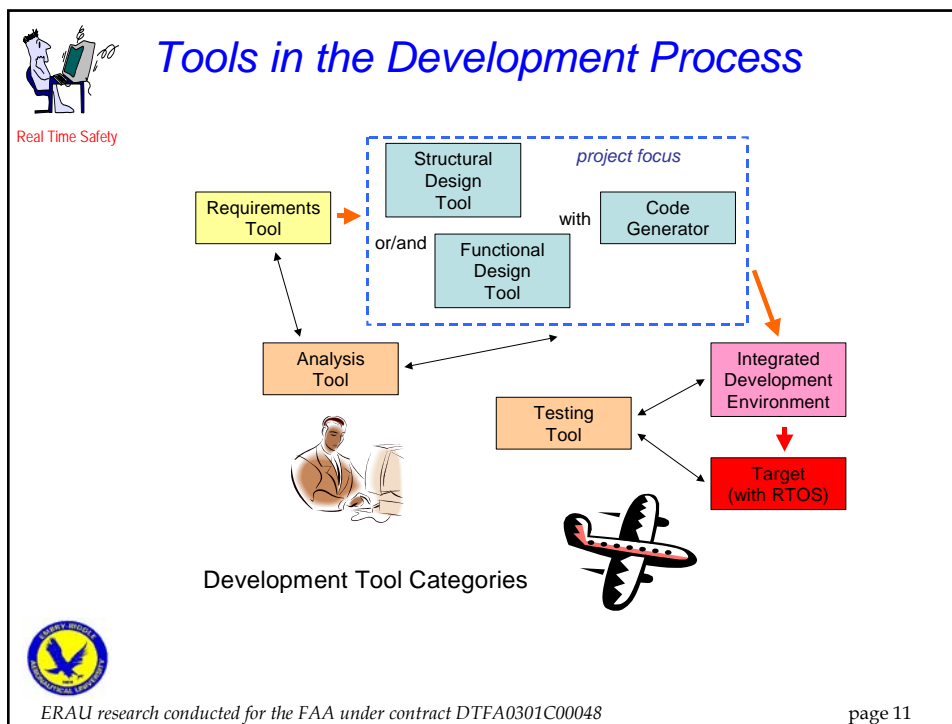


ERAU research conducted for the FAA under contract DTFA0301C00048

page 10

2003 FAA National Software Conference

Software Development Tools



2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Survey Comments

- Perspectives: applicant, certifying authority, tool vendor
- Cost, obsolescence, and inadequate documentation
- Tool functionality and compatibility with existing development environment
- Tool vendors typically not used to the level of effort required for a DO-178B compliance
- False vendor claims (tools do not scale up)
- Inadequate training/understanding of development tool
- Discouraging rigor of tool qualification and perception of qualification expensive cost
- Need for re-qualification for new certification project (reuse)
- Tool reliability as a measure?
- Vendor support and alternate means for COTS tools



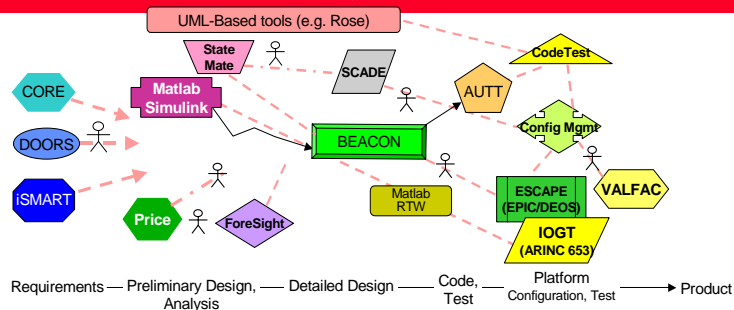
ERAU research conducted for the FAA under contract DTFA0301C00048

page 13



Real Time Safety

Current Tools: Issues, and Limitations



- Each COTS tool captures a single narrow aspect of the system design/architecture
 - ⇒ System designer must depend on other tools for architectural representation and integration
 - Custom workarounds, patches, and conversions needed around each COTS tool, in each project
- System is captured in different forms in different COTS tools
 - ⇒ Manual translation between tools causes lot of duplicated effort
- Proliferation of tools and design notations
 - ⇒ Lack of standard design/architecture notations leads to less re-use across SBUs
 - Platform dependencies get embedded in design, reducing portability



Honeywell

(courtesy of Honeywell Software Solution Lab)

ERAU research conducted for the FAA under contract DTFA0301C00048

page 14

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Qualification: Definition and the FAA View

- Qualify (*American Heritage Dictionary of the English Language*):
 - to describe by enumerating the characteristics or qualities, or
 - to declare competent or capable
- The purpose and the need of qualification (*section 12.2 of the DO-178B*):
 - “The objective of the Tool Qualification is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced or automated.”
 - “A tool may be qualified only for use on a specific system ...Use of the tool for other systems may need further qualification.”
 - “Only those functions that are used to eliminate, reduce, or automate software life cycle process activities, and whose outputs are not verified, need be qualified.”



ERAU research conducted for the FAA under contract DTFA0301C00048

page 15



Real Time Safety

Qualification: FAA View (do we need it?) (Section 9.3 of notice N8110.49)

- An affirmative answer to the following three questions is the condition for tool qualification:
 - Can the tool insert an error into the airborne software within the scope of its intended usage?
 - Will the tool's output not be verified as specified in Section 6 of DO-178B?
 - Are processes of DO-178B eliminated, reduced, or automated by the use of the tool? (i.e., will the output from the tool be used to either meet an objective or replace an objective of DO-178B, Annex A?)

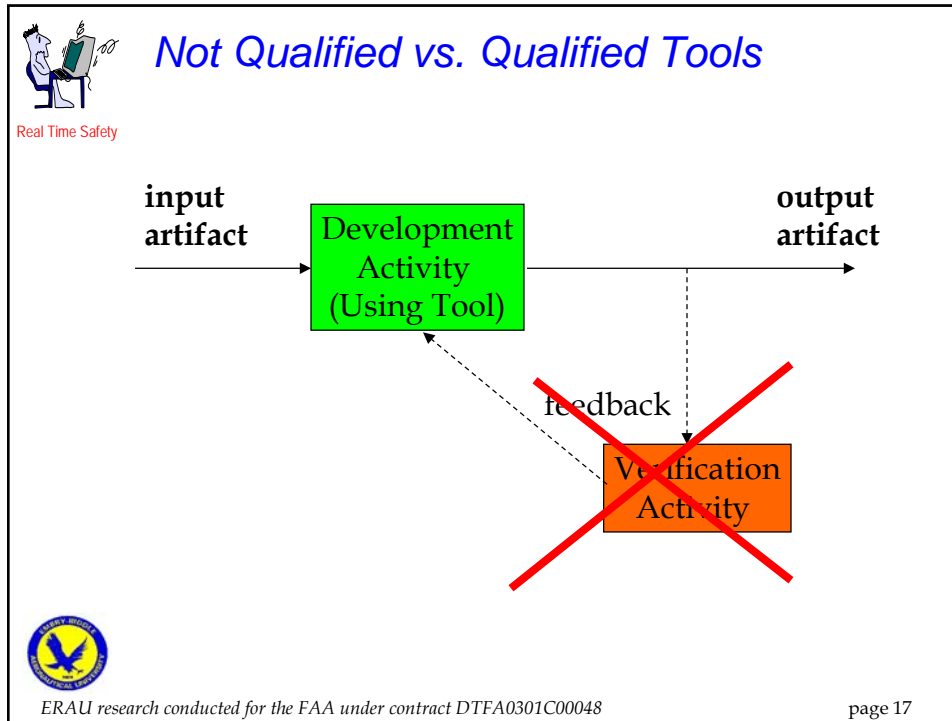



ERAU research conducted for the FAA under contract DTFA0301C00048

page 16

2003 FAA National Software Conference


Software Development Tools



 *Qualification: How is it done?*

Real Time Safety

- Tool qualifications are part of a Type Certificate (TC), Supplemental Type Certificate (STC), or Technical Standard Order (TSO) approval
- **Plan for Software Aspects of Certification (PSAC)** and **Software Accomplishment Summary (SAS)** of the original certification project need reference to **Tool Qualification Plan** and **Tool Accomplishment Summary** respectively
- Separate **Tool Operational Requirements**, **Tool Verification Records** and **Tool Qualification Development Data** need to be available
- *The requirements are described in Fig. 9.2 of the FAA Notice N8110.49 Ch.9 (old N8110.91)*



ERAU research conducted for the FAA under contract DTFA0301C00048

page 18

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Determinism: What about it?

- Restrictive interpretation of determinism: the same input necessarily leads to exactly the same output
- More accurate interpretation of determinism for tools: established the ability to determine correctness of the output from the tool
- This interpretation of determinism should apply to all tools whose output may vary beyond the control of the user, but where that variation does not adversely affect the intended use (for example, the functionality) of the output and the case for the correctness of the output is presented
- However, the generation of the final executable image should meet the restrictive interpretation of determinism



ERAU research conducted for the FAA under contract DTFA0301C00048

page 19



Real Time Safety

Determinism: Systems Classification (regarding the response to environmental stimuli)

- | ■ Interactive | ■ Reactive |
|--|---|
| <ul style="list-style-type: none">○ the system grants or allocates resources to clients on request when feasible (operating systems, data bases)○ the concerns are deadlock avoidance, fairness, data coherence○ the pace of operation is determined by computer | <ul style="list-style-type: none">○ the system reacts to external stimuli producing outputs in timely way (process control, avionics, signal processing)○ the concerns are correctness and timeliness○ the pace of operation is determined by environment |



ERAU research conducted for the FAA under contract DTFA0301C00048

page 20

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Determinism: Perspectives

- Two characteristics of typical real-time systems
 - data flows
 - state changes
- Software Engineering Viewpoint:
 - Interactive systems which use asynchronous languages based on interleaving tasks and operating systems principles (viewed as non-deterministic)
- Control Engineering Viewpoint:
 - Reactive systems using event sequencing and logical time abstraction on common discrete time scale computing one step at the time (considered to be deterministic)
 - Formal - by reducing the system to set of dynamic equations
 - Practical - by modeling and solving differential equations



ERAU research conducted for the FAA under contract DTFA0301C00048

page 21



Real Time Safety

Determinism: How to Claim It?

- Synchronous methods avoid undesirable behavioral non-determinism with programs containing parts acting concurrently but in a deterministic way
- The approach extends well-known cycle-based computation model in a form of an infinite loop repeating specific, precisely defined computation in a sequential order (making slightly unrealistic assumptions about timing relationships)
- Each block has a notion of cycle reading the input and generating output (in no time)
- State Machines have also notion of cycles, where each transition together with the associated output takes occurs in synchronous fashion



ERAU research conducted for the FAA under contract DTFA0301C00048

cont →

page 22

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Determinism: How to Claim It? (cont)

- Blocks and State Machines also communicate synchronously on a cycle basis where data tokens are exchanged without considering any change in the environment until all operations are completed
- The cycle fusion, done by the automatic code generator, considers all inter-cycle communication dependencies producing a single sequential code from the network of block diagrams and state machines
- All communication is implemented by shared variables with no context switching
- Vendors of tools based on synchronous languages (LUSTRE, SIGNAL) have made successful claims to justify the determinism and thus *qualifiability* of their tools (SCADE, Sildex)



ERAU research conducted for the FAA under contract DTFA0301C00048

page 23



Real Time Safety

Determinism: Examples of Industry Views

- Determinism is: *"... for a given model, no matter how it is constructed, it will give the same generated code and simulation answers"* and it is shown by: *"... analyzing behavior of over 6,000 models developed in the past has tested the tool only implicitly"*
- *"Qualification approach is based on the entire development strategy and on assurance that the tool complies with the tool operational requirements in the user context based on the premise that the code generated by the tool is linear and very simple - representing a sequence of calls to the library. In turn, each of the library functions is very rigorously verified."*



ERAU research conducted for the FAA under contract DTFA0301C00048

page 24

2003 FAA National Software Conference

Software Development Tools



Real Time Safety

Concerns about Airborne Software Impacting the Development Tools Qualification

- Confidence in the tool that the tool has not inserted errors into the software it produced
- The tool must guarantee the chain of correctness defined by the DO-178B
- Qualification of a tool should eliminate the manual processing of mundane and time-consuming operations
- Qualification must guarantee that the tool is predictable/deterministic
- The tool must support the implementation of large projects by supporting multi-user access, configuration management, etc.



ERAU research conducted for the FAA under contract DTFA0301C00048

page 25



Real Time Safety

Qualification vs. Assessment

- Regardless of potential qualification there must to be a way to assess the tool quality and suitability to a specific software development task
- The design phase is an essential and unavoidable part of the entire software development process
- With increased software complexity and software engineering advances, automatic development tools are more frequently used
- Because the tools contribute to the development of safety critical software, the evaluation of the tools should be made an intrinsic part of the development
- Thus, a definite process for evaluating software development tools needs to be created.




ERAU research conducted for the FAA under contract DTFA0301C00048

page 26


2003 FAA National Software Conference

Software Development Tools




Tool Assessment: Criteria and Methods

Assessment Criteria	Assessment Methods
■ Ease of validation of the tool result	■ Complexity Measures
■ Software techniques and processes used to develop the tool	■ Inspection
■ Software techniques and methodologies used by the tool	■ Compilation
■ Quality system used by the tool developer	■ Use of Standards
■ Previous use of the tool	■ Formal Methods
	■ Timing Analysis
	■ Requirement Based Testing
	■ Test Coverage Analysis
	■ Unique Identifiers
	■ Traceability
	■ Architecture Assessment
	■ Language Subset

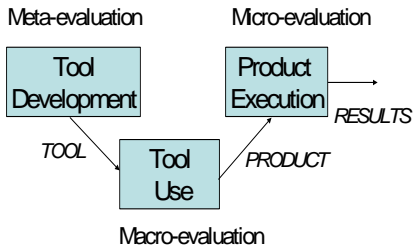


ERAU research conducted for the FAA under contract DTFA0301C00048

page 27




Tool Assessment Process: Goals



```

graph TD
    subgraph Meta-evaluation
        TD[Tool Development]
    end
    subgraph Micro-evaluation
        PE[Product Execution]
    end
    subgraph Macro-evaluation
        TU[Tool Use]
    end
    TD -- TOOL --> TU
    TU -- PRODUCT --> PE
    PE -- RESULTS --> OUT[ ]
    style OUT fill:none,stroke:none
    
```

- To help providing sufficient information to support the qualification of a tool
- To develop a set of criteria and methods to assess (measure) the quality of the tool in terms of its reliability of a functionality
- Two approaches:
 - Formal qualification-oriented evaluation of functionalities
 - Informal utilization-oriented hands-on evaluation of the tool in operation




ERAU research conducted for the FAA under contract DTFA0301C00048

page 28


2003 FAA National Software Conference

Software Development Tools




Tool Assessment Taxonomy: Approach

- Identify tool categories
- Analyze categories for their impact on aviation software development considering DO-178B guidelines
- Select categories applicable for potential qualification activities
- In each category identify functionality attributes (capabilities)
- For each capability determine:
 - Concerns
 - Factors
 - Methods



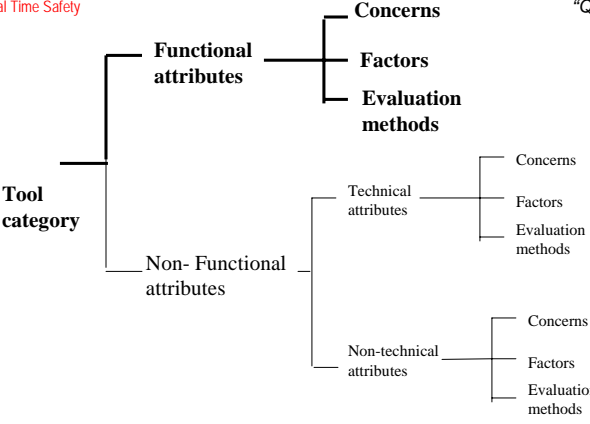
ERAU research conducted for the FAA under contract DTFA0301C00048

page 29




Taxonomy: Tool Assessment

Source: TR CMU/SEI-95-TR-021
"Quality Attributes", Barbacci, M. et al



- Technical : dependability, performance, security
- Non-technical: support, cost, vendor viability, training

- **Concerns** - user-oriented properties affecting the functionality/quality
- **Factors** - software-oriented characteristics of a concern
- **Evaluation Methods** - to assess the concerns and factors using **metrics** and **measurements**




ERAU research conducted for the FAA under contract DTFA0301C00048

page 30

2003 FAA National Software Conference


Software Development Tools



Real Time Safety


Assessment & Taxonomy: Concerns

Developer Viewpoint:	Manager Viewpoint:
<ul style="list-style-type: none">○ Functionality○ Correctness○ Accuracy○ Efficiency○ Determinism○ Traceability○ Safety○ Standards○ Documentation	<ul style="list-style-type: none">○ Cost○ Notation○ Interoperability○ Version Control○ Vendor Competency○ Vendor Reputability○ Training○ Community



ERAU research conducted for the FAA under contract DTFA0301C00048


page 31



Real Time Safety

Current Work

- For detailed analysis we selected design tools, with a **code generator** functionality, identifying two groups:
- Structural-based (UML-oriented):
 - Rose Real Time / Rational <http://rational.com/>
 - Esterel Studio / Esterel <http://esterel-technologies.com/>
 - Rhapsody / iLogix <http://ilogix.com/>
 - Real Time Studio / Artisan <http://artisansw.com/>
 - STOOD / TNI-Valiosys <http://tni-valiosys.com/>
- Functional-based (Block-oriented):
 - SCADE / Esterel <http://esterel-technologies.com/>
 - Sildex / TNI-Valiosys <http://tni-valiosys.com/>
 - Simulink/RTW / Mathworks <http://mathworks.com/products/>
 - TAU/SDL Developer / Telelogic <http://telelogic.com>
 - BEACON / Applied Dynamics International <http://adi.com>

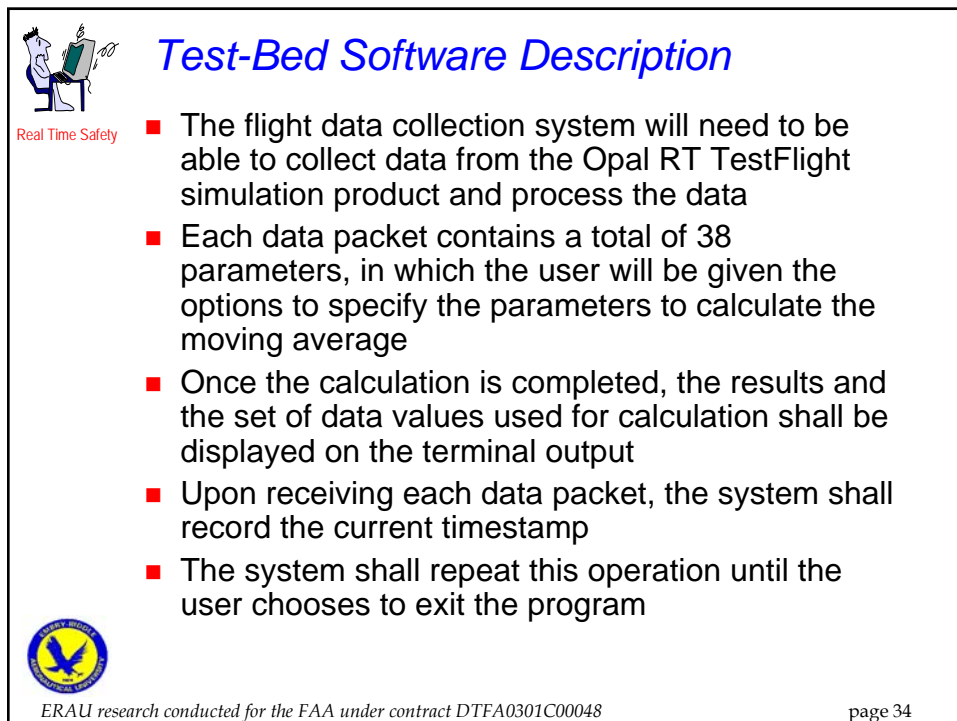
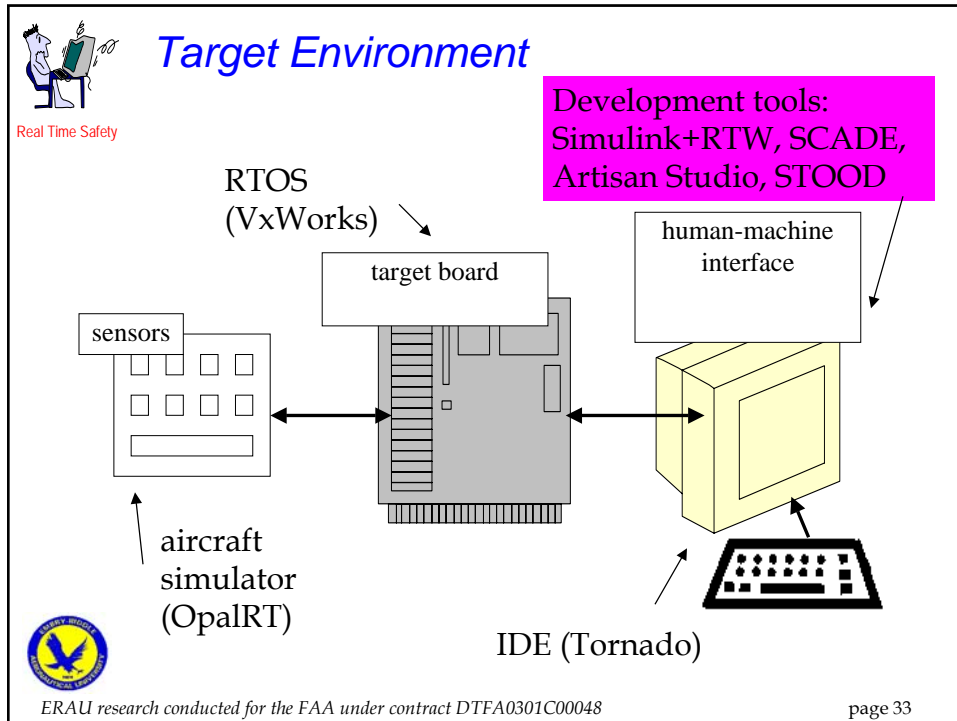


ERAU research conducted for the FAA under contract DTFA0301C00048

page 32


2003 FAA National Software Conference

Software Development Tools




2003 FAA National Software Conference

Software Development Tools




Traceability Evaluation Process (top level)

DESCRIPTION:	To perform assessment of a development tool with respect of traceability concern
ENTRY:	<ul style="list-style-type: none"> The requirements to be modeled are available
TASK	
1. Preparation	<ul style="list-style-type: none"> Prepare PSP estimates Select tool Familiarize the evaluator with the tool Familiarize the evaluator with the project
2. Model Creation and Code Generation	<ul style="list-style-type: none"> Create model representing the project Verify the model Generate source code
3. Measurement	<ul style="list-style-type: none"> Decompose model Identify code modules Identify lines of code Identify miscellaneous code
4. Postmortem	<ul style="list-style-type: none"> Complete PSP data Evaluate results and determine conclusions
EXIT:	<ul style="list-style-type: none"> The tool has been evaluated for traceability




ERAU research conducted for the FAA under contract DTFA0301C00048

page 35



Conclusion – Work in Progress

- Phase I allowed us to identify tool categories, investigate tool attributes and evaluation criteria, collect data on existing tools qualification efforts, select category of tools for detailed investigation, and prepare the environment for the case study
- Phase II continues with collecting effort and defect development process data in the case study, follow-up with the industry surveys and analyze the industry/government feedback to define tool evaluation criteria
- Phase III shall include completion of the data collection, including personnel skills when using tools, final report and the results propagation



ERAU research conducted for the FAA under contract DTFA0301C00048

page 36